

***HIGH PERFORMANCE DATABASE SERVER (HIGH AVAILABILITY
DATABASE SERVER) MENGGUNAKAN MARIADB GALERA CLUSTER***



Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada

Jurusan Informatika Fakultas Komunikasi dan Informatika

Oleh:

WIWIT WIJAYANTI

L 200 130 004

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

2020

UNIVERSITAS MUHAMMADIYAH SURABAYA

HALAMAN PERSETUJUAN

***HIGH PERFORMANCE DATABASE SERVER (HIGH AVAILABILITY
DATABASE SERVER) MENGGUNAKAN MARIADB GALERA CLUSTER***

PUBLIKASI ILMIAH

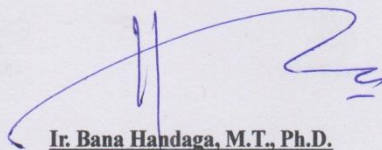
oleh:

WIWIT WIJAYANTI

L200130004

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing



Ir. Bana Handaga, M.T., Ph.D.

NIK. 793

16/4 '20

HALAMAN PENGESAHAN

***HIGH PERFORMANCE DATABASE SERVER (HIGH AVAILABILITY
DATABASE SERVER) MENGGUNAKAN MARIADB GALERA CLUSTER***

OLEH

WIWIT WIJAYANTI

L200 130 004

**Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Jum'at, 13 November 2020
dan dinyatakan telah memenuhi syarat**

Dewan Penguji:

- 1. Ir. Bana Handaga, M.T., Ph.D** (.....)
(Ketua Dewan Penguji)
- 2. Heru Supriyono, S.T., M.Sc., Ph.D** (.....)
(Anggota I Dewan Penguji)
- 3. Dedi Gunawan, S.T., M.Eng., Ph.D.** (.....)
(Anggota II Dewan Penguji)

Dekan

Fakultas Komunikasi dan Informatika



Nurgiyatna, S.T., M.Sc., Ph.D

NIK. 881

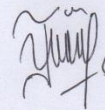
PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 13 November 2020

Penulis



WIWIT WIJAYANTI

L200 130 004



**UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA**

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

SURAT KETERANGAN LULUS PLAGIASI

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : WIWIT WIJAYANTI
NIM : **L200130004**
Judul : *High Performance Database Server (High Availability Database Server)*
Menggunakan Mariadb Galera Cluster
Program Studi : Informatika
Status : **Lulus**

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 17 November 2020

Biro Skripsi Informatika

Ihsan Cahyo Utomo, S.Kom., M.Kom.



UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

feedback studio

HIGH PERFORMANCE DATABASE SERVER (HIGH AVAILABILITY DATABASE SERVER) MENGGUNAKAN MARIADB GALER...

-- /0

< 108 of 118 > ?



HIGH PERFORMANCE DATABASE SERVER (HIGH AVAILABILITY DATABASE
SERVER) MENGGUNAKAN MARIADB GALERA CLUSTER

Abstrak

Sistem informasi merupakan bagian yang tidak dapat dipisahkan dari perkembangan teknologi saat ini. Kebutuhan akan sistem informasi yang semakin tinggi membuat layanan yang ada harus semakin lebih baik. Salah satu layanan yang menjadi pondasi dalam peningkatan kualitas sistem informasi yaitu database server. Pada sebuah jaringan atau sistem informasi yang dibangun, database server berperan sebagai tempat penyimpanan data. Kegagalan pada database server dapat terjadi kapan saja dengan sistem yang menggunakan satu server sebagai database-nya. Jika kegagalan itu terjadi, pengguna tidak dapat mengakses sistem dan berisiko kehilangan data yang disimpan di server. Oleh karena itu dibutuhkan database server yang dapat bekerja dengan cepat dan memiliki ketersediaan yang tinggi dalam performanya. Tujuan dari penelitian ini adalah untuk membangun sistem yang memiliki ketersediaan tinggi dimana beberapa database server terhubung sehingga apabila terjadi kegagalan pada salah satunya maka database server lain akan bertindak sebagai backup yang disebut cluster database server. Data pada masing-masing database server direplikasi secara real time sehingga data antar node dapat sinkronisasi. MariaDB Galera Cluster mendukung teknik replikasi database multi-master. Hasil dari penelitian ini adalah sistem yang handal dan memiliki ketersediaan tinggi.



Match Overview



22%



1 repository.unpas.ac.id 3% >
Internet Source



2 noc-id.blogspot.com 2% >
Internet Source



3 sinta3.ristekdikti.go.id 1% >
Internet Source



4 www.freelancethai.net 1% >
Internet Source



5 ejurnal.esaunggul.ac.id 1% >
Internet Source



6 nothinix.id 1% >
Internet Source

Page: 1 of 15

Word Count: 3016

Text-only Report

High Resolution

On



HIGH PERFORMANCE DATABASE SERVER (HIGH AVAILABILITY DATABASE SERVER) MENGGUNAKAN MARIADB GALERA CLUSTER

Abstrak

Sistem informasi merupakan bagian yang tidak dapat dipisahkan dari perkembangan teknologi saat ini. Kebutuhan akan sistem informasi yang semakin tinggi membuat layanan yang ada harus semakin lebih baik. Salah satu layanan yang menjadi pondasi dalam peningkatan kualitas sistem informasi yaitu database server. Pada sebuah jaringan atau sistem informasi yang dibangun, database server berperan sebagai tempat penyimpanan data. Kegagalan pada database server dapat terjadi kapan saja dengan sistem yang menggunakan satu server sebagai database-nya. Jika kegagalan itu terjadi, pengguna tidak dapat mengakses sistem dan berisiko kehilangan data yang disimpan di server. Oleh karena itu dibutuhkan database server yang dapat bekerja dengan cepat dan memiliki ketersediaan yang tinggi dalam performanya. Tujuan dari penelitian ini adalah untuk membangun sistem yang memiliki ketersediaan tinggi dimana beberapa database server terhubung sehingga apabila terjadi kegagalan pada salah satunya maka database server lain akan bertindak sebagai backup yang disebut cluster database server. Data pada masing-masing database server direplikasi secara real time sehingga data antar node dapat tersinkronisasi. MariaDB Galera Cluster mendukung teknik replikasi database multi-master. Hasil dari penelitian ini adalah sistem yang handal dengan ketersediaan tinggi.

Kata Kunci: Database Server, High Availability, MariaDB Galera Cluster, Replikasi, Sinkronisasi

Abstract

Information systems are an integral part of today's technological developments. The higher need of information systems demands the better service. One of the base foundation to improve the quality of information systems is a database server. Database server plays a role as a data storage in a network or in an information systems. A failure occurs on the database server can happen any time in a system with one server as its database. If that failure occurs, users cannot access the systems and have a risk of losing their data stored on the server. Therefore, a database server is required to work fast and has a high availability in its performance. The aim of this research is to build a high availability system where several database servers are connected so that if any failure occurs in one of them, another database server will act as a backup which is called a cluster database server. The data on each of the database server are replicated in real time so that data between nodes can be synchronized. MariaDB Galera Cluster supports multi-master database replication technique. The result of this research is a reliable system that has a high availability.

Keywords: Database Server, High Availability, MariaDB Galera Cluster, Replication, Synchronization

1. PENDAHULUAN

Sistem informasi merupakan bagian yang tidak dapat dipisahkan dari perkembangan teknologi saat ini. Kebutuhan akan sistem informasi membuat layanan yang ada harus semakin baik. Salah satu layanan yang menjadi pondasi dalam peningkatan kualitas sistem informasi yaitu *database* server. Pada sebuah jaringan atau sistem informasi yang dibangun, *database* server berperan sebagai tempat penyimpanan data (Suryanto, 2015). *Database* server menjadi bagian yang sangat penting dalam sistem informasi, dengan menggunakan *database* server penyimpanan maupun pengambilan data menjadi terstruktur dan menjadi lebih mudah.

Akan tetapi, kegagalan pada *database* server dapat terjadi kapan saja dengan sistem yang menggunakan satu server sebagai *database* nya, maka apabila terjadi kegagalan pada server tersebut maka user tidak dapat melakukan akses pada sistem informasi tersebut dan juga berisiko kehilangan data yang tersimpan pada server tersebut, oleh karena itu *database* server dituntut dapat bekerja cepat dan mempunyai ketersediaan tinggi dalam kinerjanya (Sugiyatno, 2019).

Oleh karena itu perlu adanya *database* server cadangan dalam modus siaga yang dapat secara otomatis langsung dapat online dalam beberapa saat setelah menemukan kegagalan pada server utama, hal demikian sering disebut sebagai *high availability cluster*. Selain cepat dalam melakukan penanganan saat terjadi kegagalan, konsistensi data pada setiap node sangat di perlukan, maka dari itu dibutuhkan suatu teknologi yang dapat melakukan replikasi data (Aspriyono, 2016). Replikasi merupakan suatu teknik untuk melakukan penggandaan dan pendistribusian data dari satu *database* ke *database* yang lain serta melaksanakan sinkronisasi antar *database* sehingga konsistensi data disemua *database* server terjamin (Widiono, 2019).

Pada penelitian ini, MariaDB dapat digunakan sebagai *database* server dan pengimplementasian replikasi dan *cluster database* server dapat dilakukan menggunakan MariaDB *Galera Cluster*. MariaDB *Galera Cluster* telah mendukung teknik multi-master *database replication*. Multi-master *database replication* memungkinkan seluruh node dalam satu *cluster* menjadi master *database*, sehingga jika ada perubahan yang terjadi pada satu node akan direplikasi ke seluruh node dalam *cluster* tersebut. Teknik ini dapat meningkatkan reliabilitas dan availabilitas dari sistem *database*, karena bila terjadi kerusakan atau kesalahan pada salah satu *database* server maka trafik akses *database* dapat dialihkan ke *database* server yang lain dalam *cluster* tersebut (Data, Ramadhan, & Amron, 2017).

Sehingga dalam penelitian ini akan dibangun suatu sistem yang dapat menangani apabila suatu *database* server mengalami kegagalan maka akan ada *database* server lain sebagai cadangan untuk menggantikan dan sinkronisasi data antar *database* server terjamin karena dilakukan *backup* secara *realtime* pada setiap *database* server.

2. METODE

2.1 Kebutuhan *Hardware* dan *Software*

Kebutuhan *hardware* yang digunakan dalam penelitian ini yaitu laptop Acer Aspire E1 dengan spesifikasi Core i3 i3-2348 2,3 GHz, RAM 8 GB dan *harddisk* 500 GB. Sedangkan kebutuhan *software* yang digunakan dalam penelitian ini yaitu sistem operasi server *database* yaitu ubuntu server 18.04 dimana mesin virtual yang digunakan adalah virtualbox. MariaDB digunakan sebagai *database* server dan Galera *Cluster* digunakan sebagai replikasi pada database server.

2.2 Perancangan Sistem

Penelitian ini bertujuan untuk merancang *high availability database server*, yang apabila salah satu database server mengalami kegagalan atau gangguan, maka *database* server yang lain tersedia untuk menggantikan. Dimana setiap *database* satu dan yang lainnya memiliki *backup* data yang identik (sinkron).

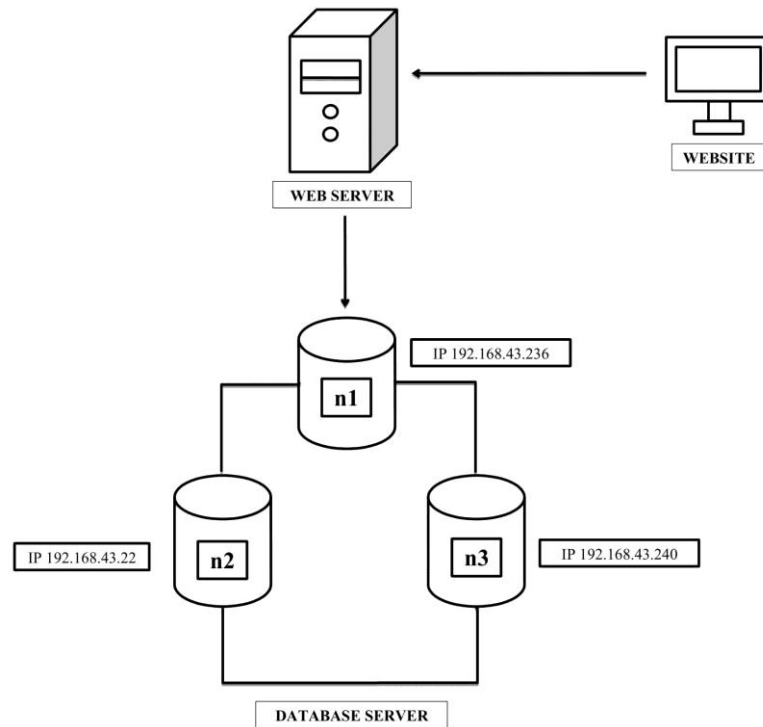
Dalam membangun perancangan sistem pada penelitian ini spesifikasi yang dibutuhkan oleh mesin virtual dijelaskan pada Tabel 1.

Tabel 1. Spesifikasi Mesin Virtual (Virtualbox)

Spesifikasi	Virtual Database Server		
	n1	n2	n3
RAM	2048 MB	2048 MB	2048 MB
Harddisk	25 GB	25 GB	25 GB
Network Adapter	Bridge Network	Bridge Network	Bridge Network
IP Address	192.168.43.236	192.168.43.22	192.168.43.240

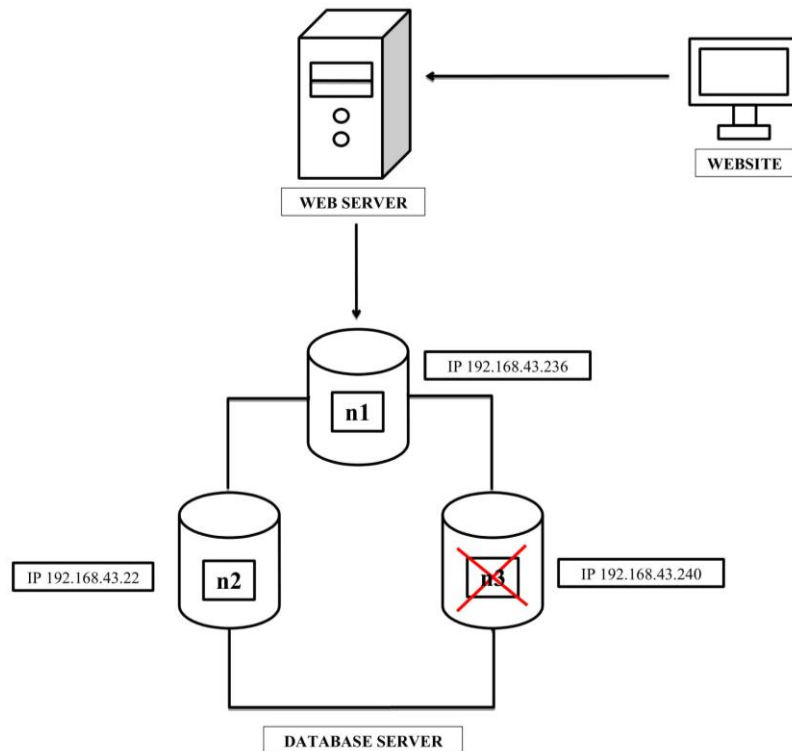
Perancangan sistem dibangun dengan menggunakan tiga buah server yang ketiganya akan diinstall MariaDB sebagai basis datanya, yang nantinya ketiga server tersebut akan saling terhubung menjadi satu kesatuan sistem *database clustering*.

Konfigurasi yang akan dilakukan untuk ketiga *database* server tersebut adalah konfigurasi *cluster* aktif-aktif, dimana semua node di dalam *cluster* bersifat baca-tulis dan perubahan yang dibuat ke salah satunya maka akan direplikasi ke semua node. Simulasi replikasi dapat dilihat pada Gambar 1.



Gambar 1. Simulasi replikasi MariaDB

Jika terjadi kegagalan pada salah satu database server maka database server yang lain akan menggantikan perannya, sehingga sistem masih dapat beroperasi seperti biasa. Simulasi kegagalan pada salah satu database server tersebut dapat dilihat pada Gambar 2.



Gambar 2. Simulasi *high availability* MariaDB Galera Cluster

2.3 Instalasi dan Konfigurasi Sistem

2.3.1 Instalasi Virtualisasi

Virtualisasi adalah cara yang digunakan untuk membuat sesuatu secara virtual atau abstrak. Virtualisasi merupakan teknik yang digunakan untuk menjalankan beberapa sistem operasi dalam satu mesin yang dijalankan secara bersamaan. Virtualisasi digunakan Virtual mesin yang digunakan sebagai media virtualisasi dalam penelitian ini yaitu virtualbox, dimana software bisa diunduh dari situs resmi <https://www.virtualbox.org>.

2.3.2 Instalasi OS Guest

Sistem operasi adalah suatu sistem yang berperan sebagai perantara antara pengguna dan perangkat keras komputer. Sistem operasi yang digunakan pada server guest yaitu ubuntu server 18.04. Ubuntu merupakan sistem operasi distribusi linux berbasis debian yang mana ubuntu juga tersedia secara gratis (*open source*). Sistem operasi ubuntu server 18.04 nantinya akan diinstall di ketiga node server.

2.3.3 Menambahkan *Link Repository* MariaDB

Sebelum melakukan penambahan *link repository* MariaDB, terlebih dahulu lakukan penambahan *repository key*, dengan perintah:

```
apt-key adv --recv-keys --keyserver  
hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8
```

Setelah *repository key* berhasil ditambahkan, kemudian baru dilakukan penambahan *link repository* MariaDB dengan perintah:

```
add-apt-repository 'deb [arch=amd64]  
http://nyc2.mirrors.digitalocean.com/mariadb/repo  
/10.4/ubuntu bionic main'
```

Setelah *link repository* MariaDB ditambahkan, lakukan *update link repository* dengan perintah:

```
apt update
```

2.3.4 Instalasi MariaDB

MariaDB merupakan DBMS yang dikembangkan oleh pengembang MySQL dimana MariaDB bersifat *open source*. Dalam perkembangannya, DBMS MariaDB sudah terintegrasi dengan MariaDB Galera Cluster.

Instalasi MariaDB dilakukan pada ketiga node, dengan mengetikkan perintah sebagai berikut:

```
apt install mariadb-server
```

2.3.5 Instalasi *rsync*

Langkah ini dimaksudkan untuk mengkonfirmasi *rsync* sudah terinstall atau memerlukan pembaharuan, dengan perintah:

```
apt install rsync
```

2.3.6 Konfigurasi Galera Cluster

Konfigurasi dilakukan untuk menghubungkan ketiga node database server agar terjadi replikasi dan sinkronisasi data. Setiap node memiliki konfigurasi yang hampir sama satu dengan yang lain. Untuk menambahkan baris konfigurasi pada setiap node, terlebih dahulu ketikkan perintah:

```
nano /etc/mysql/conf.d/galera.cnf
```

Perintah tersebut digunakan untuk mengubah atau menambahkan baris konfigurasi pada *galera.cnf*. Tambahkan beberapa baris konfigurasi yang sama disemua node, perbedaan konfigurasi hanya terdapat pada baris **#Galera Node Configuration** pada bagian **wsrep_node_address** yang diisi dengan alamat IP masing-masing node dan **wsrep_node_name** yang diisi dengan nama masing-masing node.

Konfigurasi *galera.cnf* pada n1:

```
[mysqld]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so

# Galera Cluster Configuration
wsrep_cluster_name="galera_cluster"
wsrep_cluster_address="gcomm://192.168.43.236, 192.168.43.22,
192.168.43.240"

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="192.168.43.236"
wsrep_node_name="n1"
```

Konfigurasi galera.cnf pada n2:

```
.....  
  
# Galera Node Configuration  
wsrep_node_address="192.168.43.22"  
wsrep_node_name="n2"
```

Konfigurasi galera.cnf pada n3:

```
.....  
  
# Galera Node Configuration  
wsrep_node_address="192.168.43.240"  
wsrep_node_name="n3"
```

Pengertian dari setiap baris perintah dalam konfigurasi galera yaitu pada bagian pertama dimaksudkan untuk mengubah atau menegaskan kembali pengaturan pada MariaDB yang memungkinkan cluster berfungsi dengan benar. Bagian **# Galera Provider Configuration** mengonfigurasi komponen MariaDB yang menyediakan API replikasi *WriteSet* (Galera) menentukan parameter umum untuk mengkonfigurasi lingkungan replikasi awal.

Bagian **# Galera Cluster Configuration** dimaksudkan untuk mengidentifikasi anggota cluster berdasarkan alamat IP dan membuat nama untuk cluster untuk memastikan bahwa node bergabung dengan grup yang benar. Bagian **# Galera Synchronization Configuration** menentukan bagaimana cluster akan berkomunikasi dan menyinkronkan data antar anggota. Ini digunakan hanya untuk transfer status yang terjadi saat node online dan pada bagian baris terakhir yaitu **# Galera Node Configuration** menjelaskan alamat IP dan nama server saat ini yang berguna saat mencoba untuk mendiagnosis masalah di log dan untuk mereferensikan setiap server dengan berbagai cara.

2.3.7 Membuka Firewall

Melakukan konfigurasi firewall di semua node, agar port yang dibutuhkan untuk komunikasi antar node terbuka. Port yang digunakan untuk galera adalah:

1. Port 3306 digunakan untuk koneksi klien MySQL dan *State Snapshot Transfer* (SST) yang menggunakan mysqldump.
2. Port 4567 digunakan untuk lalu lintas replikasi galera cluster. Pada port ini replikasi multicast menggunakan transport UDP dan TCP.

3. Port 4568 digunakan untuk *Incremental State Transfer*.
4. Port 4444 digunakan untuk *State Snapshot Transfer* lainnya.

Perintah untuk membuka *firewall* disetiap port adalah:

```
ufw allow 3306,4567,4568,4444/tcp
```

```
ufw allow 4567/udp
```

Melihat status *firewall* dengan perintah:

```
ufw status
```

2.3.8 Memulai Cluster Database Server

Untuk memulai cluster, terlebih dahulu matikan mariadb di ketiga node dengan perintah:

```
systemctl stop mysql
```

Setelah MariaDB pada semua node berhasil dimatikan, untuk munculkan node pertama (n1) dengan perintah:

```
galera_new_cluster
```

Kemudian untuk melihat node yang aktif pada cluster dengan perintah:

```
mysql -u root -p -e "SHOW STATUS LIKE  
'wsrep_cluster_size'"
```

Output dari perintah diatas yaitu:

```
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| wsrep_cluster_size | 1 |  
+-----+-----+
```

Gambar 3. Jumlah node yang aktif pada cluster

Untuk node kedua (n2) dan node ketiga (n3), dapat dimulai dengan perintah seperti pada umumnya, yaitu:

```
systemctl start mysql
```


3. HASIL DAN PEMBAHASAN

3.1 Pengujian Sistem

3.1.1 Pengujian Replikasi Pada Database Server

Pengujian dilakukan dengan cara membuat database baru pada n1, dengan perintah:

```
mysql -u root -p -e 'CREATE DATABASE champions';
```

Kemudian membuat table baru pada database yang sebelumnya telah dibuat, dengan perintah:

```
mysql -u root -p -e 'CREATE TABLE champions.winners (id INT NULL AUTO_INCREMENT, CLUB VARCHAR(50), TROPHY INT, COUNTRY VARCHAR(25), PRIMARY KEY (id))';
```

Menambahkan data kedalam tabel yang telah dibuat dengan perintah:

```
mysql -u root -p -e 'INSERT INTO champions.winners (CLUB, TROPHY, COUNTRY) VALUES ("Real Madrid", 13, "Spain")';
```

Setelah semua perintah dijalankan pada n1, kembali ke n2 dan n3. Bisa dilihat bahwa data yang ditambahkan pada n1 berhasil ditambahkan pada n2 dan n3 secara otomatis. Replikasi dan sinkronisasi yang dilakukan langsung pada database server berjalan dengan baik, dan data yang ditampilkan di semua node sama.

```
root@n2:/home/n2# mysql -u root -p -e 'SHOW DATABASES';
```

Enter password:

```
+-----+
| Database |
+-----+
| champions |
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+
```

```
root@n2:/home/n2# mysql -u root -p -e 'SELECT * FROM champions.winners';
```

Enter password:

```
+----+-----+-----+-----+
| id | CLUB      | TROPHY | COUNTRY |
+----+-----+-----+-----+
| 1  | Real Madrid | 13     | Spain  |
+----+-----+-----+-----+
```

(a)

```

root@n3:/home/n3# mysql -u root -p -e 'SHOW DATABASES';
Enter password:

+-----+
| Database |
+-----+
| champions |
| information_schema |
| mysql |
| performance_schema |
| test |
+-----+

root@n3:/home/n3# mysql -u root -p -e 'SELECT * FROM champions.winners';
Enter password:

+----+-----+-----+-----+
| id | CLUB       | TROPHY | COUNTRY |
+----+-----+-----+-----+
| 1  | Real Madrid | 13     | Spain   |
+----+-----+-----+-----+

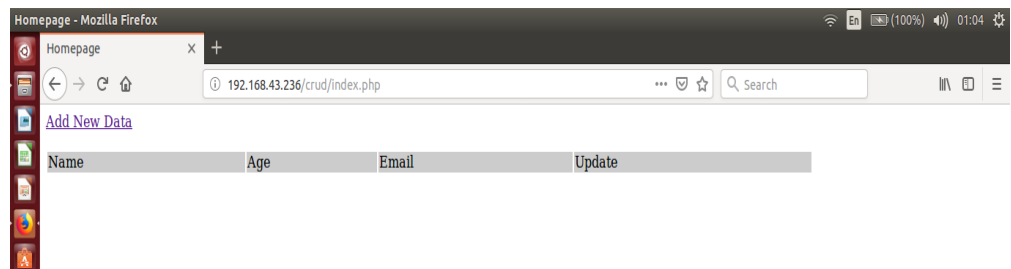
```

(b)

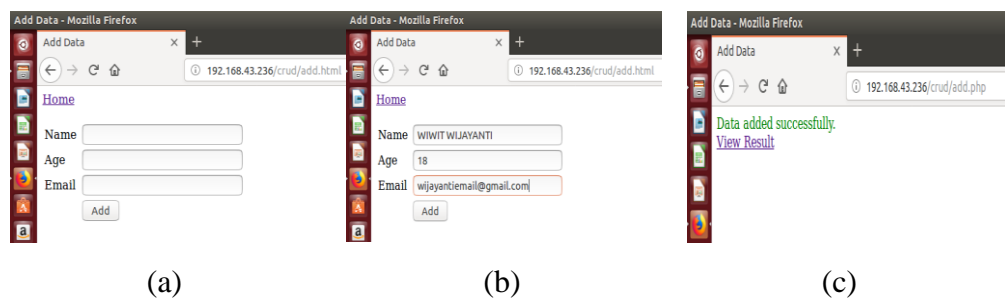
Gambar 4. (a) Menampilkan database dan tabel pada n2, (b) Menampilkan database dan tabel pada n3

3.1.2 Pengujian Replikasi Pada Website

Mengakses halaman website dilakukan dengan cara membuka browser dan mengetikkan pada alamat URL yaitu **http://192.168.43.236/crud/index.php**, dimana alamat IP yang digunakan adalah IP n1.



Gambar 5. Tampilan awal website sebelum data diinputkan

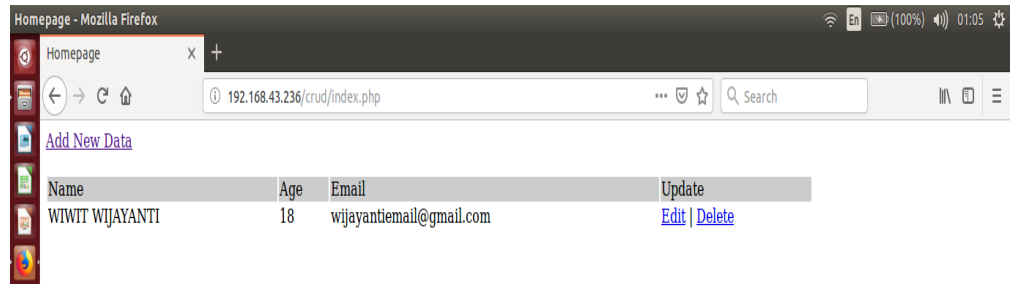


(a)

(b)

(c)

Gambar 6. (a) Halaman untuk melakukan penambahan data baru di website, (b) Menambahkan data pada website, (c) Berhasil menambahkan data



Gambar 7. Tampilan isi tabel pada website

Setelah penginputan selesai, pada database server n2 dan n3 dapat dilihat bahwa data yang diinput pada website juga ditambahkan secara otomatis pada database server n2 dan n3. Seperti pada pengujian sebelumnya yang dilakukan langsung pada database server, pengujian yang dilakukan dari website juga mendapatkan hasil yang sama, yaitu replikasi dan sinkronisasi pada setiap node telah berjalan dengan baik dan data yang ditampilkan juga sama.

```
root@n2:/home/n2# mysql -u root -p -e 'SELECT * FROM test.users';
Enter password:

+----+-----+-----+-----+
| id | name          | age | email                               |
+----+-----+-----+-----+
| 8  | WIWIT WIJAYANTI | 18  | wijayantiemail@gmail.com         |
+----+-----+-----+-----+

1 row in set (0.001 sec)
```

(a)

```
root@n3:/home/n3# mysql -u root -p -e 'SELECT * FROM test.users';
Enter password:

+----+-----+-----+-----+
| id | name          | age | email                               |
+----+-----+-----+-----+
| 8  | WIWIT WIJAYANTI | 18  | wijayantiemail@gmail.com         |
+----+-----+-----+-----+

1 row in set (0.001 sec)
```

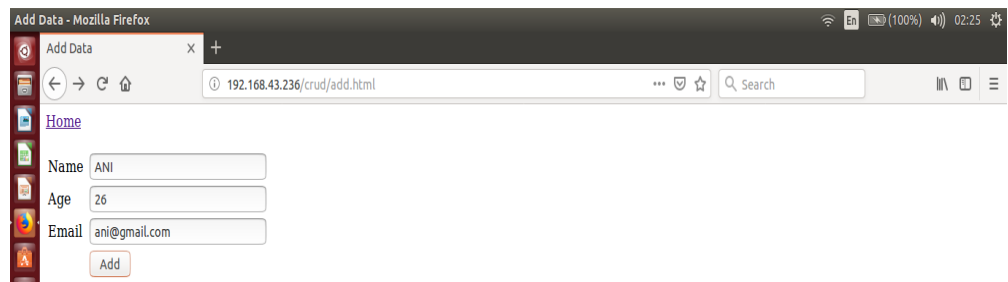
(b)

Gambar 8. (a) Menampilkan database “test” dan tabel “users” pada n2, (b) Menampilkan database “test” dan tabel “users” pada n3

3.1.3 Pengujian *High Availability*

Pengujian dilakukan dengan cara mematikan salah satu node database server. Node database server yang akan dimatikan pada pengujian ini adalah n3.

Setelah database pada n3 berhasil dimatikan, lakukan pengujian dengan cara membuka kembali website sebelumnya dan inputkan data kedalamnya.



Gambar 9. Menambahkan data baru di website pada saat n3 dimatikan

```
root@n2:/home/n2# mysql -u root -p -e 'SELECT * FROM test.users';
Enter password:

+----+-----+-----+-----+
| id | name          | age | email                               |
+----+-----+-----+-----+
| 8  | WIWIT WIJAYANTI | 18  | wijayantiemail@gmail.com          |
| 9  | ANI            | 26  | ani@gmail.com                     |
+----+-----+-----+-----+

2 row in set (0.001 sec)
```

(a)

```
root@n3:/home/n3# mysql -u root -p -e 'SELECT * FROM test.users';
Enter password:

+----+-----+-----+-----+
| id | name          | age | email                               |
+----+-----+-----+-----+
| 8  | WIWIT WIJAYANTI | 18  | wijayantiemail@gmail.com          |
| 9  | ANI            | 26  | ani@gmail.com                     |
+----+-----+-----+-----+

2 row in set (0.001 sec)
```

(b)

Gambar 10. (a) Berhasil menambahkan data pada n1 saat n3 dimatikan, (b) Berhasil menambahkan data pada n2 saat n3 dimatikan

3.2 Analisa Pengujian

Saat melakukan pengujian *high availability* database server, hanya berhasil jika yang dimatikan database pada n2 dan n3 saja, sedangkan jika database pada n1 dimatikan maka user tidak dapat mengakses atau melakukan penginputan data pada website dan koneksi antara *database* server n1 ke *database* server n2 dan n3 terputus. Jika dilakukan penambahan data langsung pada database server n3, maka data yang ditambahkan hanya akan muncul pada n2 saja sementara pada database server n1 yang dihidupkan kembali, tidak terjadi perubahan apapun dan database server akan

kembali bisa terhubung jika database n2 dan n3 dimatikan dan dihidupkan kembali. Hal tersebut terjadi dikarenakan konfigurasi IP yang dilakukan pada file config.php yang digunakan untuk mengakses website dikonfigurasi pada n1.

Maka apabila menginginkan sistem dapat tetap berjalan jika salah satu node mengalami kegagalan perlu ditambahkan teknik *failover* dan *load balancing*. Dimana teknik *failover* digunakan untuk menangani jika salah satu server mengalami kegagalan dan failover akan memindahkan layanan ke server yang lainnya. *Load balancing* merupakan teknik yang digunakan untuk pendistribusian beban trafik yang membagi beban trafik ke setiap server secara seimbang sehingga tidak ada server yang mendapatkan beban yang lebih besar.

Pada penelitian tentang *load balancing* web server dan *database cluster* oleh (Edriyansyah & Freddy, 2017) didapatkan bahwa konsep load balancing pada web server memberikan solusi dalam membangun web server yang dapat menangani permintaan *client* yang cukup tinggi dan berkualitas dan dengan adanya *load balancing* layanan web server masih dapat terus diakses oleh *client* walaupun salah satu server mengalami kegagalan karena server lain dapat menggantikan perannya sehingga seolah-olah tidak ada masalah yang terjadi pada server tersebut.

Selain itu, penelitian tentang MariaDB Galera *Cluster* menggunakan metode *load balancing* dapat dilihat pada penelitian yang dilakukan oleh (Ardiansyah, Nur, & Mukmin, 2017) yaitu dimana dua buah server digunakan sebagai penyeimbang beban (*load balancing*) dikonfigurasi aktif-pasif dan satu virtual ip digunakan sebagai *failover* ketika salah satu *load balancing* bermasalah, sehingga proses *failover* berjalan dengan baik dan tidak menghambat proses transaksi yang terjadi antara *client* dan *database cluster*.

4. PENUTUP

4.1 Kesimpulan

Berdasarkan dari penelitian yang dilakukan, dapat ditarik kesimpulan bahwa penelitian ini memiliki kelebihan dan kekurangan.

Kelebihan dari penelitian ini yaitu keberhasilan sistem dalam menerapkan replikasi dan sinkronisasi antar node yang menghasilkan data yang identik. Replikasi dan sinkronisasi yang dilakukan secara *realtime* dapat membantu dalam proses backup data yang tersimpan di dalam *database* server. Sedangkan kekurangan yang

ada pada penelitian ini yaitu *database* server pada node utama tidak boleh mengalami kegagalan, karena dapat mempengaruhi seluruh kinerja server.

4.2 Saran

Hal yang perlu diperhatikan dalam penelitian ini yaitu perancangan *clustering database* server yang harus lebih baik lagi, agar apabila *database* server utama mengalami kegagalan, maka *database* server pada node yang lain dapat menggantikan perannya.

DAFTAR PUSTAKA

- Ardiansyah, S., Nur, J., & Mukmin, M. (2017). Rancang Bangun Load Balancing Pada Database Cluster Menggunakan Haproxy. *Jurnal Informatika ISSN Online 2528-0090*.
- Aspriyono, H. (2016). Pengembangan Server Siakad Universitas Dehasen Bengkulu Menggunakan High Availability Clustering Dan MySql Database Replication. *Jurnal Ilmu Komputer*, 104-113.
- bsder. (2019, June 20). *How To Configure a Galera Cluster with MariaDB on Ubuntu 18.04 Servers*. Dipetik September 21, 2020, dari <https://www.digitalocean.com:https://www.digitalocean.com/community/tutorials/how-to-configure-a-galera-cluster-with-mariadb-on-ubuntu-18-04-servers>
- Data, M., Ramadhan, G., & Amron, K. (2017). Analisis Availabilitas dan Reliabilitas Multi-Master Database Server Dengan State Snapshot Transfers (SST) Jenis Rsync Pada MariaDB Galera Cluster. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 69-74.
- Edriyansyah, E., & Freddy, W. (2017). Rancang Bangun Load Balancing Apache Webserver Dan Database Cluster (Studi Kasus : Pada Bagian Administrasi Pembangunan Dan PDE Kota Pekanbaru). *Jurnal Ilmu Komputer Vol. 6 No. 1 E-ISSN : 2579 - 3918 / P-ISSN : 2302 - 710X*, 40-49.
- Pribadi, P. T. (2013). Implementasi High-Availability Vpn Client Pada Jaringan Komputer Fakultas Hukum Universitas Udayana. *Jurnal Ilmu Komputer*, 17-24.
- Sugiyatno. (2019). Perancangan Clustering Database Server Untuk Meningkatkan Unjuk Kerja Server Dan Menjamin Ketersediaan Layanan. *Jurnal Cendikia Vol. XVIII*, 281-289.
- Suryanto. (2015). Implementasi Clustering Database Server Menggunakan PGCluster Untuk Optimalisasi Kinerja Sistem Basis Data. *Jurnal Teknik Komputer AMIK BSI*, 134-143.

Widiono, S. (2019). Experiments and Descriptive Analysis In The Mariadb Database Cluster System To Prepare Data Availability. *International Journal of Engineering, Technology and Natural Sciences ISSN 2685-3191 Vol. 1*, 42-48.